In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [2]:

```
df = sns.load_dataset("iris")
```

In [3]:

```
df
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [4]:

```
df.head()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [5]:
```
df.tail()
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

In [6]:
```
encoder = LabelEncoder()
df["species"]= encoder.fit_transform(df["species"])
```

In [8]:
```
df.head()
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [13]:
```
df1 = df[["sepal_length","petal_length","species"]]
```

In [14]:
```
df1
```

Out[14]:

	sepal_length	petal_length	species
0	5.1	1.4	0
1	4.9	1.4	0
2	4.7	1.3	0
3	4.6	1.5	0
4	5.0	1.4	0
...	...	...	...
145	6.7	5.2	2
146	6.3	5.0	2
147	6.5	5.2	2
148	6.2	5.4	2
149	5.9	5.1	2

150 rows × 3 columns

In [15]:
```
df1.head()
```

Out[15]:

	sepal_length	petal_length	species
0	5.1	1.4	0
1	4.9	1.4	0
2	4.7	1.3	0
3	4.6	1.5	0
4	5.0	1.4	0

In [16]:
```
x=df.iloc[:,0:2]
y=df.iloc[:,-1]
```

In [18]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

In [21]:
```
clf = LogisticRegression(multi_class="multinomial")
```

In [22]:
```
clf.fit(x_train,y_train)
```

Out[22]:
```
▾            LogisticRegression
LogisticRegression(multi_class='multinomial')
```

In [23]:
```
y_pred = clf.predict(x_test)
```

In [24]:
```
print(accuracy_score(y_test,y_pred))
```

```
0.9333333333333333
```

In [25]:
```
pd.DataFrame(confusion_matrix(y_test,y_pred))
```

Out[25]:

	0	1	2
0	9	0	0
1	0	8	0
2	0	2	11

In [26]:

```
# prediction
query = np.array([[3.4,2.7]])
clf.predict_proba(query)
```

```
C:\Users\User38\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with f
eature names
  warnings.warn(
```

Out[26]:

```
array([[7.39453693e-01, 2.60361064e-01, 1.85242949e-04]])
```

In [28]:

```
clf.predict(query)
```

```
C:\Users\User38\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with f
eature names
  warnings.warn(
```

Out[28]:

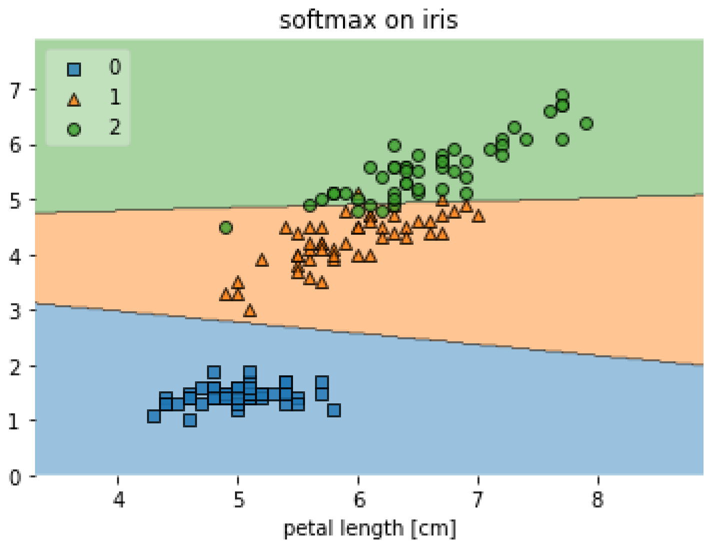
```
array([0])
```

In [33]:

```
from mlxtend.plotting import plot_decision_regions
plot_decision_regions(x.values,y.values,clf,legend=2)


# Adding axes annotations
plt.xlabel("sepal length [cm]")
plt.xlabel("petal length [cm]")
plt.title("softmax on iris")
plt.show()
```

```
C:\Users\User38\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with f
eature names
  warnings.warn(
```



In [ ]: